NFT-Based Land Ownership in Minecraft

By: Matt Lefebvre and Munir Baara

Introduction

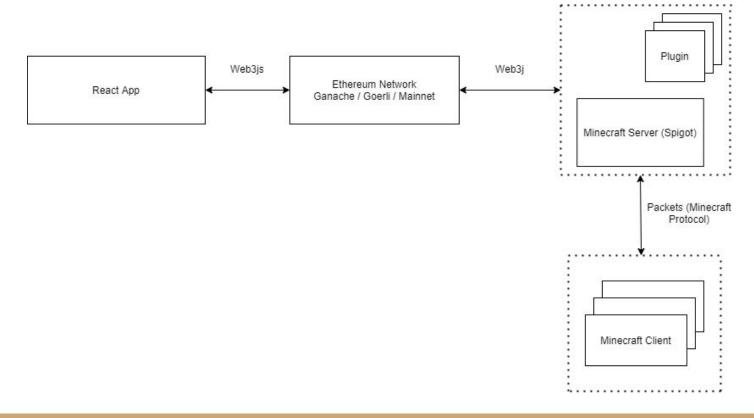
- Minecraft is a 3D sandbox game (XYZ axis)
 - X/Z axis range from [-30 million, 30 million]
 - Y axis ranges from [0, 255-384] depending on the version of the game
- A block is a single point on the XYZ axis
- A chunk is a 16x16 area that spans the Y axis
- Chunks are procedurally generated by the game
- Can do math to find out which chunk (XZ) a block (XYZ) is in



What

- NFT-Based chunk ownership done on a first come first serve basis (no whitelist, no reserved chunks, etc.)
- Players are unable to interact with blocks inside of chunks they don't own
- Three components
 - Smart contract (**Solidity**)
 - ERC721 (Non-Fungible Token Standard)
 - Each token corresponds to a chunk in-game (metadata contains the X/Z coordinate)
 - Unit tests to ensure our code works, unforgiving development environment
 - Minecraft Plugin (Java)
 - Allows users to link addresses
 - Prevents interacting with chunks the player does not own
 - Web App (React)
 - Allows users to connect MetaMask wallet and "mint"

System Design



Smart Contract

```
abstract contract ERC721Impl is IERC721, IERC721Enumerable, IERC721Metadata {
mapping(address => uint256) private _balances;
mapping(uint256 => address) private _owners;
mapping(uint256 => address) private _approvals;
mapping(address => mapping(address => bool)) private _operators;
string private _name;
string private _symbol;
uint256 private _supply;
uint256[] private _tokenIds;
```

```
contract Land is ERC721Impl {
struct Chunk {
    int128 x;
    int128 z;
}
event Mint(uint256 _tokenId, int128 x, int128 z, address owner);
mapping(uint256 => Chunk) private _tokenIdToChunk;
mapping(int256 => bool) private _claimed;
```

<<ERC721>> +Transfer(address from, address to, uint256 tokenId); <<EVENT>> +Approval(address owner, address approved, uint256 tokenId): <<EVENT> Chunk +ApprovalForAll(address owner address operator bool approved): <<EVENT>> +balanceOf(address_owner): uint256 +x: int128 +ownerOf(uint256 tokenId); address +z: int128 +safeTransferFrom(address from, address to, uint256 tokenId, bytes data): void +safeTransferFrom(address from, address to, uint256 tokenId); void +transferFrom(address _from, address _to, uint256 tokenId): void +Chunk(int128 x, int128 z); Chunk +approve(address approved, uint256 tokenId); void +setApprovalForAll(address_operator, bool_approved): void +getApproved(uint256 tokenId): address +isApprovedForAll(address _owner, address _operator): bool Land <<ERC721Enumerable>> -chunks: Chunk[] -tokenIdToChunk: mapping(uint256 => Chunk) +totalSupply(): uint256 -claimed: mapping(int256 => bool) +tokenByIndex(uint256_index): uint256 +tokenOfOwnerBvIndex(address owner, uint256 index): uint256 +Mint(uint256 tokenId. uint256 x. uint256 z. address owner): <<EVENT>> +Land(string _name, string _symbol): Land <<ERC721Metadata>> +claim(int128 x, int128 z): void +chunkBvTokenId(uint256 tokenId): Chunk +name(): string -signedIntToStr(int128 num): string) +symbol(): string +tokenUri(uint256 tokenId): string <<ERC165>> ERC721Impl +supportsInterface(bytes4 interface): bool -balance: mapping(address => uint256) -owners: mapping(uint256 => address) -approvals: mapping(uint256 => address) -operators: mapping(address => mapping(address => bool)) -supply: uint256 -name: string -symbol: string tokenIds: uint256II +ERC721Impl(string name. string _symbol): ERC721Impl mint(address to. uint256 tokenId): void

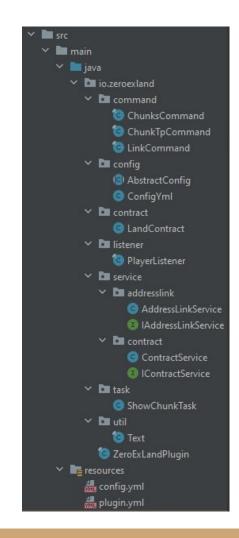
Minecraft Plugin

Commands

- /chunks (Lists owned chunks if an address is linked)
- /link <address> (Allows linking of an address)
- /chunktp <x> <z> (Go to the specified chunk)

- Listener

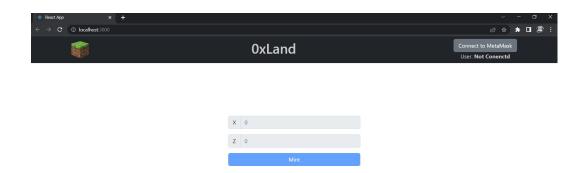
- Cancel player interacts outside of owned chunks
- Misc.
 - Configurable Web3j params (e.g. network, gas, etc.)
 - Chunk border visualization



Web App

Allow users to connect
MetaMask wallet

- Allow users to claim chunks



Future Work

- Off-chain transaction signing for account linking (Web app -> plugin)
- Dynamically create wallet file (private key) in plugin (plugin read only)

- Better UI/UX
 - Could display events emitted such as Transfer, Approval, Mint, etc.
 - Tab for owned tokens when a wallet is linked

Demo

- Contract deployed on Goerli testnet
 - Server / Web App hosted locally

Minecraft Bans NFTs

- https://www.minecraft.net/en-us/article/minecraft-and-nfts

To ensure that Minecraft players have a safe and inclusive experience, blockchain technologies are not permitted to be integrated inside our client and server applications, nor may Minecraft in-game content such as worlds, skins, persona items, or other mods, be utilized by blockchain technology to create a scarce digital asset. Our reasons follow.

End

Code: https://github.com/mlefeb01/0xLand